REMARKS

Applicant respectfully requests reconsideration of the application identified above. Claims 1-31 are pending. Claims 1, 3 and 19 are currently amended to more particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Applicants respectfully traverse the rejections as conceivably applied to the amended claims.

I. Summary of the Invention

The present invention relates to a control system for use in small buildings. The control system includes a communications network, a simple control interface and application controllers that control various small building systems, such as an HVAC, lighting, and security system. The control interface has a database populated with application controller profiles. Each of the profiles is associated with a particular application controller type. When an application controller is added to the communications network, the application controller tells the local control interface its controller type. The local controller interface is able to use its database to identify the appropriate profile to use with the application controller type. Once a profile has been identified, the control interface and application controller may communicate using the pre-defined application controller specific commands laid out in the profile.

II. Non-Art Rejection

Claims 1-31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite. In particular, the Examiner raised various antecedent basis issues. Each of the antecedent basis issues has been addressed with the current claim amendments.

In addition, the Examiner states that it is unclear what the applicants intend to cover by the term "explicit address(ing)". Applicants point the Examiner to at least page 35 lines 4-5 of the Specification, where an example of explicit addressing is given "(i.e. direct 48-bit Neuron ID addressing)." In addition, explicit addressing is a term of art that is well known in this field.

III. Art Rejections

Applicant respectfully submits that the subject matter of the amended claims is patentable over the art of record.

A. Obviousness Rejection based on Pascucci and Pouchak

As previously presented, claims 1-6, 9-18 and 29 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,550,980 to Pascucci et al. in view of U.S. Patent Application No. 2003/0005086 to Pouchak. Applicants respectfully traverse this rejection as conceivably applied to the amended claims.

Pascucci is directed to a facilities management system. As shown in Fig. 14, Pascucci teaches using a Network Control Unit (NCU) 14-1 to manage heating ventilating and air conditioning, and various other electro/mechanical systems. The Pascucci NCU includes an input/output device 14-13 and communicates with control devices 14-25 over a bus.

The Pascucci NCU is a complex device that includes various logical software levels. In particular, Pascucci discloses a feature level, a software object level and a hardware object level. Each NCU includes a set number of pre-defined features. A feature is "defined as a function performed by the system." Pascucci, Col. 56, Lns. 37-40. "The hardware object

managers format or map the data into form required by the software object managers needs. This allows the software object managers to provide a hardware independent interface to the higher level software features." Col. 37, Lns. 20-24. This means the " [software] object manager 18-27 can be written to deal with a generic object type ... [i]t need not contain alternative code for each single possible slave controller implementation of an analog input." Pascucci, Col. 36, Lns. 22-29. The device controllers in Pascucci deliver "raw data." Pascucci, Col. 35, Lns. 58-60. "The hardware object managers for the slave controllers access the node processor to condition the data and mask the differences from the [software] object manager." Pascucci, Col. 36, Lns. 1-5.

The Examiner contends that Pascucci teaches a database of a plurality of profiles. Applicant respectfully submits that the subject matter of amended claim 1 is not disclosed by Pascucci or Pouchak, either alone or in combination. Pascucci fails to teach or suggest a "plurality of profiles associated with a controller type and containing a plurality of pre-defined application controller type specific commands" as recited in amended independent claim 1. Although Pascucci teaches that "[b]efore a ... command can be issued, it must be converted to the representation understood by the particular hardware device the object controls." Pascucci, Col. 73, Lns. 9-11. No details or explanation about how this conversion is performed are disclosed. In addition, by definition, a converted generic command is not a pre-defined application controller type specific command. Accordingly, Pascucci fails to teach or suggest a "plurality of profiles ... associated with a controller type and containing a plurality of pre-defined application controller type specific commands" and Pouchak fails to fill this gap.

The Examiner cites a number of passages from Pascucci in support of his assertion that Pascucci teaches a control interface connected to said communications network, said control interface including a database of at least one profile for an application controller type. Applicants respectfully disagree with the Examiner and do not believe these passages support the Examiner's position.

The Examiner cites the following passage, which discusses the hardware object level and the software object level of the network controller.

> FIG. 18 provides a more detailed illustration of the configuration described in FIG. 17. The software object level contains a plurality of software object managers 18-27, 18-31, 18-35. A software object manager is a database manager which handles all requests for a particular type of software object. An object is a named data element stored in the network controller. Objects are categorized in types and each instance of an object has its own set of data, such as object name, current value, alarm limit, etc. Therefore, each software object manager is associated with a corresponding database, 18-29, 18-33, 18-36. One example of a software object manager is an analog input object manager 18-27. This would be the database manager for all instances of analog input objects, such as instances T1, T2, T3 of temperature objects T in the following example. Another is a Binary Input Object Manager 18-31. All of the elements of the database for a given manager are objects of the same type. In the following example, the software objects include analog input points, and binary input points. Each object type may have several instances, each of which has its own set of data values called attributes. For example, the analog input objects stored in database 18-29 are each an instance of a type of object. Each of these three instances has its own set of attributes. These attributes are data which includes the object name, the current value, the alarm limits, etc.
>
> The hardware object level contains a similar plurality of hardware object managers 18-37, 18-41, 18-45. Each hardware object manager is a database manager which handles all requests for a particular type of hardware device on the local low speed bus 17-

> 13 connecting the network controller to the slave hardware devices 17-15, 17-17, 17-19. For example, a slave controller type A hardware manager 18-37 is the database manager for all slave controller objects of type A (A1, A2, in the example given below). As with the software object managers, each hardware object manager has a database 18-39, 18-43, 18-47 associated with it containing all objects of that type. For example, database 39 for Type A hardware objects includes A1 data 18-49 and A2 data 18-51. Each object has a set of attributes unique to that particular object. Pascucci, Col. 33 Ln. 55 to Col. 34 Ln. 26

Neither the hardware nor software level described in these passages disclose a database with profiles associated with controller types of application controllers, let alone profiles with pre-defined application controller type specific commands. Instead, the databases described in the passage above include attributes such as the object name, the current value, and alarm limits.

The Examiner also points to various figures, such as Figs. 11, 15, 20 and 64, which depict network control units. Network control units do not necessarily contain databases with profiles with application controller specific commands. Accordingly, these figures do not advance the Examiner's position.

Pascucci also fails to teach or suggest "a means for configuring each of said plurality of application controllers using at least one of said plurality of pre-defined application controller type specific commands of said identified profiles" as recited in amended independent claim 1. Pascucci teaches that "[n]odes are configured on a system when they are defined in one or more storage devices as members of a network. Node configuration may occur by storing data defining a path to the node." Pascucci Col. 4, Lns. 44-47. Although Pascucci teaches configuring a node with location and path information, Pascucci does not teach configuring a

plurality of application controllers using pre-defined application controller type specific commands. Pouchak fails to fill this gap.

The Examiner cites a number of passages from Pascucci in support of his assertion that Pascucci teaches a means for configuring application controller based on a profile corresponding to a controller type. Applicant respectfully disagrees with the Examiner and does not believe these passages support the Examiner's position.

The first passage the Examiner cites discusses the addition of new controllers to the system.

> The commonality of interface between the hardware object and software object level simplifies the addition of new slave controllers and object instances. An object instance would be added in the above example if a fourth temperature sensor T4 were to be added to the system. A new slave controller of the same type would be added if a third type A slave controller, A3, were added. In both cases, all the necessary software exists on the network controller because there are no changes to the informational interfaces between the software object level 18-7 and the hardware object level 18-9. The user need only modify the database to create a new instance of the 18-29 analog input object T4 or the database 18-39 to create another instance of type A controller object, e.g. A3, in the network controller.
>
> It is also possible to add a new slave controller type with minimal impact on the existing facilities management system software. Assume a new controller type, type C, is to be attached to the local bus 17-13. This would require adding (by download or other means) a new hardware object manager to the existing software in the network controller acting as the master control node for operational units on that local bus. This hardware object manager would map the capabilities of the new controller into the software objects already defined in the system. Pascucci, Col. 36 Lns. 40-65.

Nothing in the above passage teaches or suggests a means for configuring each of the application controllers using a pre-defined application controller type specific command from a profile. Instead, this passage deals with configuring the network control unit to handle new hardware. Specifically, it teaches that 1) a *user* modifies the database on the network control unit to add the new hardware if its an existing type of controller; and 2) new software is needed if it is not an existing type of controller. There is no suggestion or teaching of using pre-defined application controller type specific commands from a database, let alone issuing such commands to configure an application controller.

The examiner also cites the following passage in support of his statement that Pascucci discloses a means for configuring an application controller based on a profile corresponding to the controller type.

> Upon downloading, an object is given to a data base manager which manages all objects of the same type. In the case of AHU1/FAN, the binary output manager in NC2 is used. The object data base manager software initially enters the names into the reference name table. For example, when the binary output object manager is given the FAN object at download time, it places the FAN object into its own binary output data base 20-2 and adds an entry in the reference table 20-300. This entry contains the name of the AHU1/FAN object and its binding information; object type (binary output), host node address (NC2), and data base location in the binary output data base (record 3). In this example, the table entry is located at offset 16 as shown in FIG. 20. Note that no other nodes besides NC2 know the binding information at this time. These nodes still only have a name reference to AHU1/FAN. The directory of objects and AHU1 will later also point to the reference table for the FAN object so that later lookups of the name will find it in the proper place. Data base for the referencing node 20-1, NC1, is also downloaded. This causes entries to be made in the reference table for NC1. When the data base for Feature A is downloaded, an entry is made into the reference table for

> AHU1/FAN but without the binding information. This "unbound" reference simply shows that some feature in NC1 will be referencing FAN. When the data base for high level software Feature B is downloaded, it will also try to add the unbound reference to the table and find that it is already there (at offset 27 in the reference table 20-302). Both features will then replace the named reference to AHU1/FAN with the offset of 27 into the table. Note that at this point in time the binding information is still not in the table in NC1. Col. 42 Lns. 8-35 of Pascucci.

The database described in this passage does not contain profiles with pre-defined application controller type specific commands. The database includes the name of the object, its binding information, object type, host node address, and database location in the binary output data base. There is no suggestion or teaching of pre-defined application controller type specific commands in a profile within a database, let alone issuing such commands to configure an application controller.

B. Anticipation Rejection based on Pascucci

As previously presented, claims 19-27 were rejected under 35 U.S.C. § 102(b) as being unpatentable over U.S. Patent No. 5,550,980 to Pascucci et al. Applicants respectfully traverse this rejection as conceivably applied to the amended claims.

Applicant respectfully submits that the subject matter of amended claim 19 is not disclosed by Pascucci. Pascucci fails to teach or suggest a "pre-defined application controller specific commands" as recited in amended independent claim 19. As noted above, Pascucci does teach that "[b]efore a ... command can be issued, it must be converted to the representation understood by the particular hardware device the object controls." Pascucci, Col. 73, Lns. 9-11. However, Pascucci does not teach any details about how this conversion is performed.

Furthermore, at best, the Pascucci commands are converted generic command and not pre-defined application controller specific command as required by amended independent claim 19. Accordingly, Pascucci fails to teach or suggest a "pre-defined application controller specific commands".

Pascucci also fails to teach or suggest a "means for adjusting a value said plurality of control variables in accordance with said pre-defined application controller specific commands" as claimed in independent claim 19. As noted above, Pascucci teaches that "[n]odes are configured on a system when they are defined in one or more storage devices as members of a network. Node configuration may occur by storing data defining a path to the node." Pascucci Col. 4, Lns. 44-47. Pascucci does not teach adjusting control variables in accordance with pre-defined application controller specific commands.

C. Dependent Claims

The dependent claims recite additional subject matter not present in the corresponding independent claims, these dependent claims are even more clearly allowable over the art of record than the corresponding independent claims. Claim 2 discloses application controllers that control operation of corresponding automated device in accordance with at least one variable. Claims 3 and 20 disclose a plurality of profiles with different controller types. Claim 4 discloses transmitting explicit messages to said application controllers as opposed to Pascucci which appears to use peer to peer or implicit messages. Claim 5 discloses incorporating the value0 of the variable into the explicit message. Claim 6 discloses a database that includes input, output, and configuration data structures for application controllers. Claims 7 and 30

disclose application controllers with occupancy status. Claims 8 and 23 disclose a network server interface. Claims 9 and 24 include a means for monitoring including a means for periodically transmitting a ping to each application controller. Claims 10 and 25 disclose a means for receiving a ping from the control interface and a means for transmitting a response. Claims 11 and 22 disclose that the application controllers includes a HVAC., lighting or access control application. Claim 12 discloses a control interface with a database of application control software images. Claim 13 discloses downloading the software images onto the control interface from an external source. Claim 14 discloses an application controller being programmed to receive and install software images downloaded by the control interface. Claim 15 discloses means for downloading a control interface control software image. Claim 16 discloses a means for downloading the control software image from an external source. Claim 17 discloses a control interface programmed to receive and install a downloaded control software image. Claim 18 discloses a control interface and application controllers programmed with a generic programming language. Claim 21 discloses a preprogrammed database with a plurality of profiles that are uniquely associated with one of the controller types. Claim 27 discloses the response transmitted by an application controller includes data relevant to another application controller. Claim 28 discloses a means for generating an alarm if an application controller fails to respond to a ping. Claim 29 discloses a self-configuration means. Claim 31 discloses a means for calculating a person count and a means for defining an occupancy status within the group based on the person count.

IV. Conclusion

It is respectfully submitted that the subject matter of the amended claims is not anticipated by the art of record and that any attempt to reconstruct the subject matter of the amended claims through a combination of prior art references can only be made in hindsight with the present invention as a blueprint. However, even such an improper combination does not teach or suggest the present invention for the reasons noted above. It is therefore respectfully submitted that the rejection under 35 U.S.C. §§ 102(b) and 103 are unfounded or overcome, and therefore should be withdrawn.

In view of the above amendments and remarks, it is respectfully submitted that the present application is in condition for allowance. A notice to that effect is earnestly and respectfully requested.

Respectfully submitted,

CHRISTOPHER KIKTA ET AL

By: Warner Norcross & Judd LLP

/Dustin H. Bettendorf/
Dustin H. Bettendorf
Registration No. 56,507
900 Fifth Third Center
111 Lyon Street, N.W.
Grand Rapids, MI 49503-2487
(616) 752-2197

DHB
1401286-2